SELECT, INSERT, UPDATE y DELETE en SQL: aprende a utilizar cada uno

Giulia Losnak
16/06/2021

Un amigo mío que trabaja en una biblioteca me pidió que creara un sistema para que los empleados hicieran un seguimiento del stock de libros, qué libros se tomaron prestados, cuántos estaban disponibles, cuándo se agregó un nuevo libro al inventario, entre otras cosas.

Sugerí hacer un sistema en la base de datos, para que pudieran hacerlo fácilmente y tener su propio sistema, en el que pudieran tener este control.

Empecé descargando el **MySQL**, configuré un servidor y creé **CREATE** una`database biblioteca`, con el comando:

```
CREATE database biblioteca;
```

Ahora que se ha creado la base de datos, que es la propia base de datos de la biblioteca, debemos agregar los datos de los libros. ¿Y cómo podemos hacerlo?

En SQL, los datos se insertan en tablas, en las que cada columna representa datos y cada línea, un libro registrado.

Creando tablas con MySQL

Empecé a pensar en lo que tendría que estar en la tabla. Llegué a los siguientes tópicos:

- Título
- Autor
- ISBN
- Edición
- Editorial
- Año de publicación
- Páginas
- Género literario
- Idioma
- Número de libros en esta ediciónNúmero de libros de esta edición disponibles en la biblioteca

Sin embargo, recordé que antes de comenzar a crear algo en la base de datos, debemos **informar qué base de datos queremos usar**. Para esto usamos el comando **USE** y el nombre de la base de datos cada vez que revisamos esa base de datos. Entonces escribí:

```
USE biblioteca;
```

A partir de ahí, comencé a escribir el código para crear CREATE la tabla TABLE y el nombre de la tala, que será libros. Resultando:

```
CREATE TABLE libros
```

Para colocar las columnas, debemos abrir y cerrar un paréntesis y escribir las columnas entre ellos, colocando lo que es cada uno.

Previo el nombre de la columna, debemos poner el tipo de información que recibirá.

Para empezar, en la columna de título y autor, cómo serán textos, y a menudo pueden ser grandes, puse la categoría **text** que no necesita establecer un límite de caracteres. Entonces quedó:

```
 CREATE TABLE libros (
```

```
titulo text,

autor text,
```

Sin embargo, en las columnas de ISBN, edición y editorial, que son informaciones que pueden contener algún número o carácter especial, la coloco en la categoría **varchar**, y como no podemos dejar una gran posibilidad, pongo un límite de 50 caracteres cada uno. Resultando:

```
ISBN varchar (50),
edicion varchar (50),
editor varchar (50),
```

Entonces, creé la columna del año para indicar la fecha en la que se publicó el libro, siendo la categoría **year**, que acepta solo el año, por lo que no es necesario incluir la fecha completa:

```
añoPublicacion year,
```

Para la columna de páginas, como esta información siempre será un número entero, pongo la categoría **int** que indica que solo se agregarán números enteros.

```
ctdPaginas int,
```

En la columna de género, como siempre insertamos los mismos tipos, puse la categoría **enum**, que solo le permite ingresar un valor que se encuentra entre algunos que se definieron al crear la tabla.

Así, también podemos saber cuántos libros de un determinado género tenemos en stock. Y quedó de la siguiente manera:

```
genero ENUM ('poesia', 'soneto', 'romance', 'fábula', 'novela', 'cronica', 'cu
```

En la categoría de idiomas, también puse varchar porque podemos recibir libros en diferentes idiomas. Resultando:

```
idioma varchar(50),
```

Por último, coloqué las columnas de la cantidad total de ejemplares de ese libro disponibles en la biblioteca y la cantidad que está disponible, que se debe actualizar cada vez que se toma prestado un libro.

```
CREATE TABLE libros (

titulo text,

autor text,

ISBN varchar (50),

edicion varchar (50),

editorial varchar(50),

añoPublicacion year,

ctdPaginas int,

genero ENUM ('poesia', 'soneto', 'romance', 'fábula', 'novela', 'cronica', 'cu

idioma varchar(50),
```

Más columnas y clave primaria

La biblioteca quiere una columna para la cantidad total y para la cantidad disponible para que una esté siempre actualizada, mientras que la otra permanece con el mismo valor mientras recibamos un libro, para tener un control de cuántos libros deberíamos tener - y para saber si alguno de ellos se ha perdido.

Como solo pueden ser números enteros, ya que no podemos tener medio libro, puse en la categoría int:

```
cantidad int,

disponible int );
```

Además, al final del insert, luego de cerrar el paréntesis, debemos terminar el comando con un punto y coma (;), como deberíamos hacer al final de cada comando en SQL.

El código entero para crear la tabla se ve así:

```
CREATE TABLE libros (
titulo text,
autor text,
ISBN varchar (50),
edicion varchar (50),
editorial varchar(50),
añoPublicacion year,
ctdPaginas int,
genero ENUM ('poesia', 'soneto', 'romance', 'fábula', 'novela', 'cronica', 'cu
idioma varchar(50),
cantidad INT,
disponible int);
```

Antes de ejecutar, pensé en crear una columna que en la primera línea tuviera el valor “1” y fuera aumentando los valores automáticamente para que este número indicara cuántos libros tenía la biblioteca en total.

Pensando en cómo hacer esto, recordé que en la **Apostilla de SQL de Alura** se creaba una columna "id" que se incrementaría automáticamente con el comando auto_increment y sería una **clave primaria o PRIMARY KEY**, que es una columna que tiene un valor único en cada línea, que los identifica.

```
CREATE TABLE libros (
titulo text,
autor text,
ISBN varchar (50),
edicion varchar (50),
```

```
editorial varchar(50),
añoPublicacion year,
ctdPaginas int,
genero ENUM ('poesia', 'soneto', 'romance', 'fábula', 'novela', 'cronica', 'cu
idioma varchar(50),
cantidad int,
disponible int );
```

¡Listo! ¡Cree mi tabla!

Ahora, con la tabla creada, ¡tenemos que agregarle datos! ¿Y cómo lo podemos hacer?

INSERT: insertando datos en la tabla

Para esto, usé el comando **INSERT**

Para usar INSERT debemos escribir **INSERT INTO** y el nombre de la tabla. Luego coloque entre paréntesis las columnas que tendrán un valor insertado, escriba **VALUES** y escriba en otro paréntesis los valores que se insertarán en las columnas.

Dado que la columna id se incrementará automáticamente cada vez que se agrega una línea, no es necesario insertar un valor en ella.

Entonces, agregaremos una línea, que son los datos que tendrán en la tabla:

```
INSERT INTO libros (titulo, autor, isbn, edicion, editorial, añoPublicacion, c
```

Sin embargo, apareció el siguiente error:

```
Error Code: 1136. Column count doesn’t match value count at row 1
```

Cuando busqué para saber más, descubrí que este error ocurre cuando colocamos valores en columnas que no existen. Sin embargo, todas estas columnas existen en mi tabla. Cual será el problema?

Al analizar el comando, noté que no había ningún valor ingresado en la columna idioma, aun haber informado en el comando que recibiría un valor. Reintentando:

```
INSERT INTO libros (titulo, autor, isbn, edicion, editorial, añoPublicacion, c
```

¡Y funcionó! Entonces, comencé a agregar los otros artículos de ese mes.

Sin embargo, ¿cómo puedo hacer para mostrar la tabla en la pantalla? Pensé que debería seleccionar toda la tabla o una parte de ella para que se mostrara, por lo que el comando es **SELECT**:

```
SELECT ALL FROM libros;
```

Sin embargo, presentó el siguiente error:

```
Error 1064. You have an error in your SQL syntax.
```

Esto significa que entendí mal la sintaxis SQL, es decir, que el comando o una parte de él, estaba mal escrito. Fue cuando busqué y encontré que en SQL el comando para seleccionar todo es *, así que puse:

```
SELECT * FROM libros
```

¡Y aparecieron todas las cosas que registré en la base de datos!

id	titulo	autor	isbn	edicao
1	orgulho e preconceito	jane austen	978-8544001820	luxo
2	orgulho e preconceito	jane austen	978-8544001820	luxo
3	1984	george orwell	978-8535914849	1ª
4	a hora da estrela	clarice lispector	978-8532508126	1ª
5	dom casmurro	machado de assis	978-8525406798	bolso
6	a redoma de vidro	sylvia plath	978-8525057945	1ª
7	admirável mundo novo	aldous huxley	9788525046611	1ª
8	o peso do pássaro morto	aline bei	9788569020233	1ª
9	a rosa do povo	carlos drummond de andrade	9788501025975	1ª
10	a rosa do povo	carlos drummond de andrade	9788501025975	1ª

DELETE – Eliminando líneas de tablas

Sin embargo, al mirar la lista, noté que había agregado la misma edición dos veces en la portada de Orgullo y Prejuicio.

Como la cuestión es controlar la cantidad de título que tenemos, no es necesario tenerlo dos veces, cuando la cantidad de libros está en la tabla, ¿verdad? ¿Y cómo puedo borrar una de las líneas duplicadas?

Para borrar en la computadora usamos **DELETE**, entonces podemos usar el mismo comando:

```
DELETE from libros
```

Sin embargo, me di cuenta de que no había indicado qué línea debería eliminarse y usando el comando, así que **podría borrar toda la tabla**, es decir, todos los datos que había introducido en ella.

Entonces, pensé en especificar qué línea tenía al menos un valor en una de las columnas. En mi tabla, ¿hay una columna que tenga un valor único para cada línea?

¿Recuerda que cuando creamos la tabla creamos la columna id que tendría un número que se incrementaría automáticamente y que era la clave primaria de la tabla?

¡Esta es la columna que deberíamos usar como parámetro! Porque, si usáramos la columna de nombre, las dos líneas quedarían eliminadas y tendríamos que volver a incluirlas. Usando la columna id, que tiene un número diferente para cada línea, solo se elimina la línea incorrecta.

Entonces, usé nuevamente SELECT * FROM libros y vi en la tabla la id de la línea duplicada.

id	titulo	autor	isbn	edicao
1	orgulho e preconceito	jane austen	978-8544001820	luxo
2	orgulho e preconceito	jane austen	978-8544001820	luxo
3	1984	george orwell	978-8535914849	1ª
4	a hora da estrela	clarice lispector	978-8532508126	1ª
5	dom casmurro	machado de assis	978-8525406798	bolso
6	a redoma de vidro	sylvia plath	978-8525057945	1ª
7	admirável mundo novo	aldous huxley	9788525046611	1ª
8	o peso do pássaro morto	aline bei	9788569020233	1ª
9	a rosa do povo	carlos drummond de andrade	9788501025975	1ª
10	a rosa do povo	carlos drummond de andrade	9788501025975	1ª

En este caso, la id es 2.

Cambiando el comando para eliminar la línea de la tabla, donde la id era 2, quedó:

```
DELETE from libros WHERE id=2;
```

¡Y funcionó! Volviendo a ver la tabla, no hay más línea 2 y solo hay una línea para esta edición de Orgullo y Prejuicio.

id	titulo	autor	isbn
1	orgulho e preconceito	jane austen	978-8544001820
3	1984	george orwell	978-8535914849
4	a hora da estrela	clarice lispector	978-8532508126
5	dom casmurro	machado de assis	978-8525406798
6	a redoma de vidro	sylvia plath	978-8525057945
7	admirável mundo novo	aldous huxley	9788525046611
8	o peso do pássaro morto	aline bei	9788569020233

¡Y pronto! ¡Ahora la biblioteca empezó a funcionar!

UPDATE - Cambiando datos de la tabla

Los libros ya están siendo prestados y ¿cómo puedo actualizar la cantidad de libros disponibles en la biblioteca?

Para cambiar los datos de una columna en una línea de una tabla en SQL, use el comando **UPDATE**. Sin embargo, al igual que DELETE, debe especificarse para que no se modifiquen todas las líneas.

Estaba escribiendo el comando para actualizar la línea de la columna, usando el nombre del libro como parámetro.

```
UPDATE libros SET disponible=1 WHERE titulo='El Diario de Anne Frank';
```

Sin embargo, tenemos varias ediciones de este mismo libro y usando este comando, cambiaríamos la cantidad de libros disponibles para todas las ediciones. Entonces, necesitamos una información de la tabla que sea única en cada línea. ¿Cuál sería?

¡Sí! ID de la línea, que se incrementa automáticamente y tiene un número específico para cada línea, nos permite saber exactamente qué línea queremos cambiar.

Entonces, comencé a cambiar los comandos, usando la id que es la clave primaria de la tabla para cada línea como parámetro de verificación de la línea. Como aquí:

```
UPDATE libros SET disponible=1 WHERE id=19;
```

o peso do pássaro m...	aline bei	1	1
a rosa do povo	carlos d...	2	2
harry potter e a câm...	jk rowling	3	3
harry potter e o prisi...	jk rowling	3	3
harry potter e o cáli...	jk rowling	3	3
harry potter e a ord...	jk rowling	3	3
harry potter e o eni...	jk rowling	3	3
harry potter e as relí...	jk rowling	3	3
harry potter e as relí...	jk rowling	3	3
harry potter e as relí...	jk rowling	3	3
o diário de anne frank	anne frank	2	1

¡Funcionó! ¡Y así podemos actualizar la cantidad de libros disponibles para esa edición ¡cada vez que uno se toma prestado y se devuelve!

SELECT - Mostrando datos de la tabla

Después de un tiempo con la biblioteca en funcionamiento, una persona preguntó cuántos libros de poesía tiene la biblioteca, que tenían ediciones disponibles para pedir prestado. ¿Y cómo podríamos conocer estos datos para responderle?

Vimos anteriormente cómo mostrar todos los datos en la tabla, que es con `SELECT * from libros`.

Pero cómo podemos usar este comando para mostrar solo los títulos que son poesía y no todos los libros que tiene la biblioteca, ¿cómo funcionará el comando si lo ejecutamos así?

Para ello, necesitamos utilizar el título del libro como parámetro para que podamos seleccionar solo las ediciones de ese mismo título.

Entonces, siguiendo la idea de `DELETE` y `UPDATE`, donde debemos mostrar qué líneas van a verse afectadas, haremos lo mismo con `SELECT`.

```
SELECT * from liBros WHERE genero='poesia' AND disponible>0;
```

¡Y funcionó! Sólo aparecen los títulos de poesía E que tienen ediciones disponibles para pedir prestadas.

id	titulo	autor	isbn	edicao
8	o peso do pássaro morto	aline bei	9788569020233	1ª
9	a rosa do povo	carlos drummond de andrade	9788501025975	1ª

Sin embargo, así aparecen todas las informaciones del libro y quiero saber solo el título, el autor, el género y la cantidad disponible, ¿verdad? ¿Cómo puedo usar el comando SELECT para mostrar solo estas columnas?

Probé el siguiente comando:

¡Y funcionó! Sólo aparecieron las líneas que son del género poesía.

titulo	autor	disponivel
o peso do pássaro morto	aline bei	1
a rosa do povo	carlos drummond de andrade	2

Ahora que sabemos cómo crear una tabla, insertar, eliminar, cambiar y seleccionar datos de ella, ¿qué tabla puedes crear con la base de datos? Accede a nuestros **cursos de SQL** y descubre ¡qué más puedes hacer con base de datos!

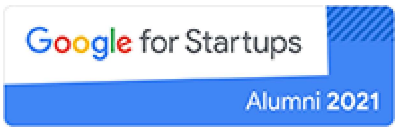
Puedes leer también:

- ¿Qué es SQL?
-
- select count(*), count(1) y count(nombre): batalla de los counts de SQL

ARTÍCULOS DE TECNOLOGÍA > DATA SCIENCE

ALIADOS

En Alura somos unas de las Scale-Ups seleccionadas por Endeavor, programa de aceleración de las empresas que más crecen en el país.



Fuimos unas de las 7 startups seleccionadas por Google For Startups en participar del programa Growth Academy en 2021

POWERED BY

CURSOS

Cursos de Programación
Lógica de Programación | Java

Cursos de Front End
HTML y CSS | JavaScript | React

Cursos de Data Science
Data Science | Machine Learning | Excel | Base de Datos | Data Visualization | Estadística

Cursos de DevOps
Docker | Linux

Cursos de Innovación y Gestión